

Kapittel 3 - Resonnering ved bruk av caser

Formålet med kapittel 3

- Gi mer detalj om hvilken type slutninger caser kan støtte
 - Tilfeller der man kan utnytte case-basert "inference"
 - Effekten av disse slutningene på forskjellige resonneringsproblemer

Resonnering ved bruk av caser

- Omfanget av resonneringsproblemer som caser støtter kan deles inn i to kategorier:
 - Problemløsningsoppgaver (design, planlegging og diagnose)
 - Tolkingsoppgaver (forståelse, begrunnelse og projeksjon)
- Resonnering ved bruk av caser har en tendens til å være forskjellige for problemløsningsoppgaver og tolkningsoppgaver
 - I problemløsning så foreslåes det løsninger (ballpark solutions), som igjen modifiseres for å gi en løsning
 - I tolkningsoppgaver så ser man på likheter og kontraster for å få en forståelse av en ny situasjon basert på kjente ting

Resonnering ved bruk av caser

- Caser kan brukes på en stor variasjon av slutningsoppgaver og resonnementsoppgaver, men ikke for alle
 - Inneholder mye rådata som kan føre til forskjellig måter å gjøre slutninger gitt omstendighetene
 - Et stort problem knyttet til å ta slutninger er å vite "hvilken type når?"
- Noen domener er så godt kjent at man trenger kun noen regler for å gi et godt resonnement
- Andre er så lite kjent at kun casene gir grunnlag for en slutning

Case-baserte slutninger

- MEDIATOR demonstrerte variasjonen av slutningsproblemer som case-baserte slutninger støtter
- MEDIATOR hadde som oppgave å løse disputer
 - Forstå problemet godt (politisk eller økonomisk)
 - Foreslå en løsning ved å generere en plan
 - Evaluere resultatene, og fikse feil
- Ved å bruke essensielt de samme case-baserte slutningsprosessene for oppgaver involvert i å forstå, planlegge og fikse feil, så demonstreres den brede bruken for case-baserte slutninger.
- MEDIATOR viser også at case-baserte slutninger fungerer bra der en lang resonneringskjede er nødvendig for å gi en konklusjon, der man kun trenger å sjekke en case for konsistens, istedet for å generere en fra scratch

CBR og problemløsning

- Å løse problemer, forstå situasjoner og andre oppgaver vi vil resonnerer oss fram til, krever en kjede av slutningssteg.
- Ved å bruke caser så blir denne kjeden mye kortere
 - Det foreslåes nesten riktige caser
 - Gir advarsel om potensielle feiltagelser

CBR for planlegging

- Planlegging er en prosess for å finne en sekvens av steg for å oppnå en tilstand i et domene (Tower of Hanoi)
- CHEF, som lager oppskrifter, er et eksempel på en case-basert planlegger.
- Problemer innenfor planlegging er blant annet:
 - Senere steg i en plan skal ikke "angre" (undo) tidligere steg
 - En planlegger må sørge for at alle forhåndskrav er oppfylt
 - Case-baserte planleggere håndterer disse problemene ved at den gir planer som allerede har blitt brukt, der disse problemene er blitt løst.
- Har man flere mål som skal nåes, kan man hente inn tidligere caser som er indeksert av konjunksjonen av mål de oppnår. (CHEF)

CBR for planlegging

- Andre kompleksiteter er å håndtere sammenhengen imellom planlegging og utføring på en realistisk måte
 - Man kan ikke forvente at en sekvens av steg virker i den virkelige verden
 - Det er urealistisk å tro at man vet alt i starten - ofte finner man ut ting etter hvert
 - Verden er uforutsigbar
- Det er derfor nødvendig at en planlegger er i stand til å gjøre "execution-time repairs"
- PLEXUS som vet hvordan man kjører undergrunnsbanen er i stand til dette (nye måter å kjøpe billetter på, osv)
- MEDIC som planlegger steg for å diagnostisere lungesykdommer (kan plutselig prioritere andre ting)
- Flere eksempler på dette i 3.2.1

CBR for design

- I designproblemer så er problemer definert som begrensingsproblemer
- Ofte flere løsninger (underconstrained)
 - Middagsplanleggeren JULIA, har ofte flere mulige løsninger tilgjengelig
- Noen ganger kan det ikke oppfylles (overconstrained)
 - Da må man prøve å oppfylle de viktigste problemene
 - I megling må man ofte finne et kompromiss
- Design-caser gir illustrasjoner på hvordan flere begrensninger har blitt håndtert i tidligere løsninger
- Arkitekter og ingeniører ser ofte igjennom mange tidligere caser før de starter å lage en løsning
- Å løse et problem ved å tilpasse en gammel løsning gjør at man unngår å håndtere mange begrensninger

CBR for design

- I nesten alle designproblemer så trenger man flere caser for å løse et problem
 - Store
 - Bruke et case som rammeverk - fyller inn med andre
 - JULIA bruker kompositt-caser for å lage et rammeverk (forret, hovedrett og dessert)

CBR for forklaring og diagnostisering

- Problemet er å finne årsaken til at noe skjer
- En case-basert tilnærming til problemet er å forklare et fenomen ved å hente fram et tilsvarende fenomen.
- Eks: Hesten som dør i et løp:
 - Jim Fixx (hjertefeil) - krever få slutninger
 - Janis Joplin (dop) - krever flere slutninger
- Diagnostisering er et forklaringsproblem der man har et sett med symptomer.
 - Eks: Diagnostiserer en pasient med depresjon, men pasienten har også vondt i magen.
 - Husker en tilsvarende case, der diagnosen viste seg å være noe annet

CBR for tolkning

- En vanlig måte å evaluere situasjoner er å huske liknende gamle situasjoner og se på likheter og kontraster.
 - Eks: En søker på en skole har gode karakterer, men viser seg å være veldig umoden.
- Høyestrettsdommere bruker CBR for tolkning
 - Likheter/ulikheter for en case
- Case-basert tolkning er mest nyttig i tilfeller hvor man ikke har en beregningsmetode for å evaluere en løsning eller posisjon
 - Selv om en slik metode hadde vært tilgjengelig, så er det ofte altfor mye ukjent til at utregninger lar seg gjøre

CBR for tolkning - "Justification and Adversial reasoning"

- Adversial reasoning vil si å gi argumenter for å overbevise andre at vi har rett
 - Noe vi gjør daglig mot andre
 - Også mot oss selv
 - Ofte vil man trenge gamle opplevelser og caser for å støtte opp om argumenter
- Juss er et godt domene for dette

CBR for tolkning og klassifikasjon

- Tolkning i CBR vil ofte si å avgjøre om et konsept passer inn i en klassifikasjon.
- Mange klassifikasjoner som vi antar er veldefinerte, kan ofte vise seg å være "åpne"
 - Antar at et kjøretøy er en ting med hjul som brukes til transport.
 - Med "kjøretøy forbudt" så omfatter dette muligens ikke rullestoler eller barnevogner, selv om de passer definisjonen
- En case-basert klassifiserer spør om den nye instansen er lik nok en annen, slik at det kan klassifiseres likt.
- Hvis ingen caser passer, så vil det noen ganger være nødvendig å bruke hypotetiske caser.
 - HYPO gjør dette ved hjelp av heuristikker

Tolkende CBR og problemløsning: Projeksjon

- Det klareste eksempelet på nyttigheten av tolkende metoder innenfor problemløsning, er projeksjon av effekter/resultater av foreslåtte løsninger.
- Projeksjon:
 - Prosessen der man forutser effekter av avgjørelser eller planer
 - Når alt om en situasjon er kjent, kan man kjøre kjente slutninger fra en løsning og se hvor det leder hen
 - Ofte er ikke alt kjent, og man kan ikke forutse effekter baser på enkle "inference"-regler
 - En av de viktigste flaskehalsene innenfor planlegging

Tolkende CBR og problemløsning: Projeksjon

- Eks: Kommandøren som finner en strategi før et slag
 - Har en doktrine med generelle regler, for å lage en enkel plan
 - Mye uvisse faktorer
 - Motstanderens strategi
 - Om det er lett å legge feller i området
 - En god plan må uansett lages, og den bør bli evaluert basert på forutsette resultater
 - Virket liknende planer, eller ikke?

Case-basert og andre resonneringsmetoder

- Positive ting med CBR:
 - Trenger ikke å finne løsninger fra scratch
 - Advare mot potensielle problemer
 - Slippe å bruke mye tid på resonnering
 - Det at gamle løsninger virket, gjør at man i stor grad slipper å argumentere for liknende løsninger
 - Gjør det ofte bedre enn andre metoder
- Ulemper:
 - Utforsker ikke fullt ut løsningsrommet, noe som gjør at man potensielt aldri vil finne den beste løsningen
 - Noe som også er sant for andre metoder som bruker heuristi

kker

- Case-basen kan fort ta stor plass

CBR og regel-basert resonnering

- RBR: Regler i regelebaser er "patterns"
 - CBR: Caser i case-baser er konstanter
- RBR: Man henter regler som matcher inputen akkurat
 - CBR: Man henter caser som passer delvis
- RBR: Regler brukes i en iterativ sykel av "microevents"
 - CBR: Caser hentes først, slik at hele løsningen tilnærmes med en gang, deretter så formes den for å passe en løsning
- RBR: Regler er små, ideelt sett uavhengige, men konsistente deler av domenekunnskapen
 - CBR: Caser er store deler av domenekunnskapen, høyest sannsynlig delvis redundant sammenliknet med andre caser
- Det har vist seg at man bygger et case-basert system tilsvarende et regel-basert system mye raskere

CBR og model-basert resonnering

- Begge har som mål å unngå resonnering fra scratch
- Begge deler kunnskap inn i store deler
 - Hovedforskjellen er hva denne kunnskapen inneholder
- Kunnskapsinnhold:
 - MBR: lagrer kausale modeller av enheter eller domener
 - CBR: lagrer caser som beskriver hvordan ting fungerer
- Domenebruksområde:
 - MBR: Kan brukes når en kausal modell eksisterer, dvs, når et domene er godt nok kjent til å kunne lage en kausal modell
 - CBR: Kan brukes under det samme som MBR, men også tilfeller der domenet ikke er godt kjent.

Case-basert- og model-basert resonnering

- Oppgaveanvendbarhet
 - MBR: brukes for å verifisere løsninger, men løsningsgenereringen er "unguidet".
 - CBR: brukes for effektiv løsningsgenerering og evalueringen baserer seg på de beste casene som er tilgjengelig.